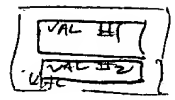
each value assembly line receiving input value packages via the input interfaces;

combining the input value packages in each value assembly line in accordance with rules defined in a specific main line function, making a value contribution, and generating a value added package;

in each value assembly line, making the value added package available via the output interface;

wherein similar types of information are contained in the input value packages and in the value added package;

providing an uppermost hierarchy level having precisely one value assembly line of the highest hierarchy level;



therefore, $UHL = VAL_{HHL}$

the precisely one value assembly line generating a value added package;

the precisely one value assembly line of the highest hierarchy level receiving input value packages via its input interfaces from lower-order value assembly lines;

feedback?

representing a lower-order value assembly line in an entirely similar way as a value assembly line which receives its input value packages from lower-order value assembly lines and whose value added package is provided as input value package for the value assembly line of the uppermost hierarchy level; and

for at least one of each hierarchy level down to a lowermost hierarchy level, representing a lower-order value assembly line of the at least one hierarchy level in a similar way as a value assembly line of the next lower hierarchy level, which likewise receives input value packages via input interfaces, combines received input value packages,

each VAL is represented the same

makes a value contribution, and makes a value added package available at the output interface;

representing the value assembly process as a fractal process in the case of which the structure of all value assembly lines is similar on all hierarchy levels, value packages being processed in accordance with the following steps:

on a lowermost hierarchy level, supplying value packages to the value assembly lines of the lowermost hierarchy level across the system boundaries of the value assembly process;

combining the input value packages of a lowermost hierarchy level in value assembly lines of the lowermost hierarchy level in accordance with its main line function, increasing the value of the value packages by a value contribution of the value assembly line and/or of the main line function, and making a value added package available at the output interface of the value assembly line;

on all hierarchy levels up to a highest hierarchy level, passing the value added package on to precisely one value assembly line of the next higher hierarchy level, as an input value package;

wherein the value flows take place strictly in one direction, in each case from a lower hierarchy level into a higher hierarchy level, and the value assembly lines of a hierarchy level are not interconnected.

2. The method as claimed in claim 1, comprising:

comparing each value added package with a reference value added package;

and detecting and reporting impermissible deviations of the value added package and of the reference value added package via a warning function.

3. A computer program for simulating and illustrating a value assembly process, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which value assembly process value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated, the computer program comprising:

machine-readable instruction sequences of a first, higher-order hierarchy level, which prompt a computer to read in data from at least one data form and to combine these data using specific combining rules, and to store the output data thus determined in a data form on a computer-readable storage medium and/or an output medium;

mutually independent machine-readable instruction sequences of at least one further hierarchy level of lower order than the first hierarchy level, which instruction sequences prompt a computer to read out data at least from a number of data forms, to combine them with one another, and to store the results of combination in an output data form of this hierarchy level in each case; and

machine-readable instruction sequences which prompt a computer to read data into input data forms from an input unit;

wherein the input data forms, which are being read on a specific hierarchy level during the program run, are output data forms of a hierarchy level of lower order than this hierarchy level, or input data forms;

wherein all data forms have a standardized data format in such a way that all output data forms which are generated during the run of instruction sequences on an arbitrary hierarchy level, and all input data forms have a common data structure, that is to say data which are of one information type are always stored in the same position in a form;

and wherein all different machine-readable instruction sequences of all different hierarchy levels are generated from identical source codes from which instructions for reading in the specific data forms and relating to the specific combinations are created by a specific parameter file either during generation of the sequences which can be executed, or during the running time of the computer program.

4. The computer program as claimed in claim 3, wherein the data contained in the forms constitute input value packages and value added packages of the value assembly process.

5. The computer program as claimed in claim 3, wherein the data generated by an instruction sequence are classified according to their qualitative information content and stored in different classes of standardized data forms.

6. The computer program as claimed in claim 3, wherein during execution of the computer program or individual sequences of the computer program the generated value added packages are compared with a reference value added package in each case, which reference value added package contains specification data, and wherein a report is made via a warning function in the event of impermissible specification deviations.

7. The computer program as claimed in claim 3, wherein machine-readable sequences of different value assembly subprocesses run on different computers, and data forms are transferred via long-distance data lines.

8. A method for visually representing a value assembly process on an output unit of a computer system, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, wherein value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages are combined in each value assembly line, a value contribution

is made, and a value added package is generated, the computer system comprising a central processing unit and a pointing device in addition to the output unit, the method comprising:

representing a value assembly line of one hierarchy level on the output unit, said value assembly line being represented as an arrow at the tip of which a value added package is transferred, and representing lower-order value assembly lines which are value assembly lines of a lower-order hierarchy level, as arrows the tips of which are applied to the shaft of the arrow which represents the value assembly line of the higher-order hierarchy level;

a user of the computer system selecting a lower-order value assembly line via the pointing device;

representing the selected lower-order value assembly line in the same way as the value assembly line of the higher-order hierarchy level as an arrow with smaller arrows running up to it.

9. The method as claimed in claim 8, wherein a computer program in accordance with claim 3 runs in the central processing unit of the computer system.

10. The method as claimed in claim 8, wherein the arrows of value assembly processes which make different value contributions are represented with the aid of different colors and/or line thicknesses.

11. The method as claimed in claim 8, in which a computer program runs in the central processing unit of the computer system, and in which the warning function prompts the computer system to display on the output unit an impermissible value deviation occurring on an arbitrary hierarchy level, and all arrows which represent value assembly lines which are affected by the value deviation are represented in a particularly emphatic type of visual representation on the output unit in such a way that the impermissible deviation can be traced back immediately to its origin from a higher hierarchy level.

15. The method of claim 8, comprising:

in response to the user selecting the arrow tip of the value assembly line via the pointing device, representing the value assembly line of the higher-order hierarchy level in a similar way to the value assembly line, to allow the path of an integral part of the value assembly process to be traced back without difficulty to its origin from a higher hierarchy level, and to allow the contribution of this integral part to the overall value assembly process to be traced through all hierarchy levels.